As described in ~~copending application Serial No. 09/478,571~~ U.S. Patent No. 6,665,726, which is also incorporated herein by reference, live streaming can be further enhanced by having the CDN send multiple copies of the same stream over different routes from a CDN entry point to the optimal streaming server at the edge of the Internet. These copies are then combined to form one complete, original-quality stream, which is sent from the streaming server to the end users. Figure 2 illustrates this process in more detail. A broadcast stream 200 is sent to a CDN entry point 202. An entry point, for example, comprises two servers (for redundancy), and each server can handle many streams from multiple content providers. Once the entry point receives the stream, it rebroadcasts copies of the stream to set reflectors 204a-n. The streams are multiplexed and delivered to the set reflectors preferably via UDP (e.g., WMT encapsulated in RTSP encapsulated in UDP over IP). These set reflectors are preferably diverse from a network and geographic standpoint (e.g., at diverse Internet backbone data centers) to ensure fault tolerance. Each set reflector, in turn, rebroadcasts its copy of the stream to each subscribing region, e.g., region 206d, of a set of regions 206a-n. A subscribing region 206d is a CDN region that contains one or more streaming edge nodes 208a-n to which user(s) have been routed by the CDN request-routing mechanism. In other words, set reflectors send their streams to every edge region where they are needed. A CDN region, in this example, includes a set of edge nodes connected by a common backbone 209, e.g., a local area network (LAN). Typically, an edge node, e.g., node 208d, comprises a streaming server 212 and it may include a cache 210. A representative server runs an Intel processor, the Linux operating system and a Real Media or QuickTime Server. For Windows-based platforms, a representative server runs an Intel processor, Windows NT or 2000, and a Windows Media Server. As will be described, the edge node also runs control programs 214 to facilitate the inventive subscription mechanism."

On page 8, beginning at line 27, please delete the paragraph that begins and page 9, line 22, replace "Serial No. 09/478,571" with "U.S. Patent No. 6,665,726.

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 9, line 19 and continuing through page 10, line 1 and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

Each subscribing region, then, simultaneously receives multiple copies of the streamed content. These copies have been sent via separate routes over the Internet, so congestion resulting in dropped packets is unlikely to impact each copy of the stream equally. As described in ~~copending Serial No. 09/478,571~~ U.S. Patent No. 6,665,726, each region preferably has a mechanism to recreate in real time an original version of the stream as sent to the entry point. In this way, the technique compensates for the inherently faulty Internet and inherently lossy UDP transport protocol. The reassembly mechanism within each region makes the original, verbatim stream available to every streaming media server within that region. When a user clicks on a CDN-tagged stream, the stream is delivered from the optimal edge node (and, in particular, that node's streaming media server) identified by the CDN's request-routing mechanism. If the CDN maps a user to a node in a region which has not subscribed to that broadcast stream (which, for example, is true for the first connection served from that region), the region automatically notifies the set reflectors and subscribes to that stream."

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 13, line 29 and continuing through page 14, line 20 and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

If the decision process determines that another server is a better source than a current source, control is passed to the stream switch process 310, which is the process that makes the actual switch from one server to another. The particular technique used for switching from a first server to a second server typically is media type-dependent and any convenient technique may be used. Thus, for example, assume a stream being received has a length of 5:00 minutes and the client player is processing the stream from a first server at an offset of 2:35 when a decision is made to change to a second server. The stream switch process 310 may then cause the second server to begin sending the stream at an offset of 2:45 and, at the same time, send appropriate instructions to terminate the stream from the first server. The stream switch process thus has the capability of picking the server it likes "least" and instructing that server to cease transmission at a given point. In another alternative, once the stream switch process 310 decides to make a switch, it creates an internal buffer and causes that buffer to be filled with advance portions of the stream (e.g., by instructing the first server to deliver packets faster than those packets can be rendered or by instructing the player to slow down the rendering process). The switch process 310 then causes the first server to cease transmission and request that the new server begin the stream at a given offset. At this point, the pre-cached data in the buffer is rendered until the given offset is reached and the new data is received. As appropriate, the stream switch process includes the capability to match data packets from first and second servers to enable a substantially seamless switch to the new stream source. Voice streams can be synched before a switch by snipping out and putting in minute bits of silence between words. Of course, the above are merely exemplary as any convenient switching technique may be implemented. ~~(DAVID, you can describe a few others if appropriate)~~